Mitigação de Envenenamento de Rótulos em Sistemas de Detecção de DDoS Federados

Pedro H. Barros[1], **Fabricio Murai**[2], **Amir Houmansadr**[3], **Antonio A. F. Loureiro**[1], **Alejandro C. Frery**[4], **Heitor S. Ramos**[1]

[1] Universidade Federal de Minas Gerais – Brazil

[2]Worcester Polytechnic Institute, USA

[3]University of Massachusetts Amherst, USA

[4]Victoria University of Wellington, New Zealand

pedro.barros@dcc.ufmg.br

Abstract. *Network traffic monitoring is essential for understanding infrastructure behavior and assessing component integrity. Federated learning has emerged as a promising approach for defense systems based on traffic monitoring, enabling distributed model training without direct data sharing. However, traditional methods assume a federated environment composed solely of honest clients, overlooking the possibility of label poisoning attacks. This paper proposes a novel federated learning framework robust against network attacks, focusing on mitigating malicious clients. Our approach employs Siamese Networks techniques to quantify data adherence and dynamically adjust the weighting of each client's contributions, enhancing the model's resilience against adversarial manipulations. The results demonstrate that our strategy not only improves attack detection but also significantly reduces the impact of label poisoning on federated learning.*

Resumo. *O monitoramento de tráfego de rede é essencial para compreender o comportamento da infraestrutura e avaliar a integridade de seus componentes. O aprendizado federado tem se destacado como uma abordagem promissora para sistemas de defesa baseados nesse monitoramento, permitindo o treinamento distribuído de modelos sem compartilhamento direto de dados. No entanto, métodos tradicionais assumem um ambiente federado composto apenas por clientes honestos, ignorando a possibilidade de ataques de envenenamento de rótulos (label poisoning). Este trabalho propõe um novo arcabouço de aprendizado federado robusto contra ataques de rede, com foco na mitigação de clientes maliciosos. Nossa abordagem emprega técnicas de Redes Siamesas para quantificar a aderência dos dados e ajustar dinamicamente a ponderação das contribuições de cada cliente, fortalecendo a resiliência do modelo contra manipulações adversárias. Os resultados mostram que nossa estratégia não apenas melhora a detecção de ataques, mas também reduz significativamente o impacto de envenenamento de rótulos no aprendizado federado.*

1. Introdução

Nos últimos anos, o número de dispositivos conectados à Internet cresceu significativamente. No entanto, a proliferação de dispositivos com baixo poder computacional

trouxe novas vulnerabilidades de segurança, como evidenciado pelo botnet Mirai, que explorou a insegurança de dispositivos IoT [Li et al. 2020]. Esses dispositivos são alvos atrativos para ataques devido à sua ampla disponibilidade e menor segurança em relação a servidores na nuvem. Nesse cenário, métodos de Sistemas de Detecção de Intrusão (IDS) tornam-se essenciais para monitoramento e defesa de redes. No entanto, abordagens tradicionais, como a detecção baseada em assinaturas, têm limitações contra ataques novos [Barros et al. 2022]. Modelos de aprendizado de máquina (ML) podem superar esses desafios, adaptando-se a novas ameaças e padrões de tráfego malicioso [Lavaur et al. 2022].

O Aprendizado Federado (FL) surge como uma solução promissora ao permitir treinamento colaborativo de modelos de ML em dispositivos distribuídos, preservando a privacidade dos dados [Li et al. 2020]. Diferentemente de abordagens centralizadas, no FL os dados permanecem localmente nos dispositivos, enquanto apenas os modelos treinados são compartilhados, garantindo menor latência e maior eficiência no uso de largura de banda. Entretanto, ataques em ambientes federados apresentam desafios únicos devido à colaboração entre múltiplas organizações com diferentes políticas de segurança. Um exemplo crítico são os ataques DDoS, que podem indisponibilizar serviços, causar prejuízos financeiros e comprometer a reputação das organizações [Li et al. 2022a]. Em um ambiente federado, um ataque pode afetar não apenas a organização alvo, mas também as demais participantes da rede compartilhada.

Embora existam trabalhos sobre detecção de ataques DDoS em FL na literatura, muitos assumem cenários idealizados, como clientes honestos ou dados completamente rotulados [Chen et al. 2022]. Para superar essas limitações, propomos um novo framework distribuído e escalável para detecção de ataques em redes, robusto a ataques de envenenamento de rótulos.

Nossa proposta introduz uma abordagem de agregação federada baseada em um quantificador de assertividade do modelo, permitindo diferenciar contribuições de clientes honestos e maliciosos, aumentando a resiliência do modelo contra ataques. A eficácia do framework é avaliada em dois conjuntos de dados reais, demonstrando um desempenho superior às técnicas existentes, com redução significativa de erro em cenários com clientes maliciosos na federação. As principais contribuições deste trabalho são: (i) um modelo supervisionado para detecção de ataques em ambientes federados; e (ii) evidências de que o modelo é eficaz mesmo com apenas 5% de dados rotulados.

2. Trabalhos Relacionados

Questões de segurança em sistemas de aprendizado de máquina, incluindo FL, têm sido amplamente estudadas [Wang et al. 2020, Su and Qu 2022]. [Li et al. 2021b] propõem um sistema federado para detecção de intrusão, usando vetores prototípicos para correlacionar espaços de representação locais e o modelo global. [Tian et al. 2021] utilizam redes neurais residuais, enquanto [Lv et al. 2022] propõem redes convolucionais, ambos alcançando alta precisão na detecção de ataques DDoS no conjunto CICDDoS2019. [Zhang et al. 2021] analisam cenários com dados não-IID, utilizando um algoritmo hierárquico de agregação baseado em K-Means e SMOTEENN, alcançando F1-Scores de até $99,62\%$. Outros estudos focam em arquiteturas baseadas em computação em névoa. [Li et al. 2022a] utiliza FL para mitigar ataques, enquanto [Dao et al. 2022b] introduzem

o FOGshield, um sistema que utiliza mapas auto-organizáveis para filtrar tráfego anômalo. Por outro lado, [Toldinas et al. 2022] transformam fluxos de pacotes de redes em representações visuais para classificação em cenários FL, alcançando F1-Scores superiores a 93%.

Embora técnicas não supervisionadas sejam promissoras [Li et al. 2022a, Dao et al. 2022b], algoritmos supervisionados frequentemente oferecem maior desempenho [Van Engelen and Hoos 2020]. [Liu et al. 2023] utilizam LSTMs bidirecionais para detectar DDoS com 5% dos dados rotulados, enquanto [Yin et al. 2022] integram FL com blockchain para detecção confiável de ataques multi-domínio, atingindo mais de 95% de precisão em diversas categorias.

2.1. Discussão

A maioria dos métodos citados indica a viabilidade de identificar pacotes de DDoS em ambientes de FL. A Tabela 1 resume as abordagens de FL relacionadas ao tema discutido neste artigo. A tabela está organizada em colunas que descrevem diferentes aspectos-chave das abordagens: se a proposta foi treinada em um cenário de aprendizado supervisionado, se a federação continha clientes maliciosos, se o modelo foi treinado com amostras pequenas de dados e se a proposta é robusta contra ataques de envenenamento de rótulos. As células são marcadas com um "✓" (i.e., sim) ou um "✗" (i.e., não), respectivamente.

Modelos supervisionados em FL oferecem benefícios como maior precisão e menor necessidade de dados, mas também introduzem novas vulnerabilidades, como ataques de envenenamento de rótulos, que podem comprometer a integridade do modelo [Nguyen et al. 2021]. Apesar de várias abordagens supervisionadas para detecção de DDoS [Zhao et al. 2019, Li et al. 2021b, Tian et al. 2021, Zhang et al. 2021, Chen et al. 2022, Su and Qu 2022, Lv et al. 2022, Neto et al. 2022, Ali et al. 2023, Liu et al. 2023], a literatura tipicamente não considera essa nova superfície de ataque. Para mitigar esse risco, é essencial adotar medidas de segurança, como validação e criptografia de dados [Nguyen et al. 2021]. O uso de blockchain pode auxiliar na seleção de participantes confiáveis e prevenir ataques de envenenamento [Yin et al. 2022], porém sua implementação ainda apresenta desafios relacionados a custo e complexidade [Issa et al. 2023].

A obtenção de grandes quantidades de dados rotulados é um desafio devido a restrições de tempo e recursos. Uma alternativa viável é o uso de pequenas amostras de dados rotulados para o treinamento de modelos, o que pode melhorar a eficiência na detecção de ataques DDoS. Na literatura, [Liu et al. 2023] abordam essa estratégia, mas sem considerar ataques ao modelo na federação. Diferente disso, nosso trabalho explora a vulnerabilidade de modelos federados, propondo uma abordagem que considera tanto ataques à rede quanto ao modelo, mitigando os impactos do envenenamento de modelos em FL.

Tabela 1. Resumo das abordagens discutidas na Seção 2.

Trabalho	Aprend. Sup.	Cliente Malic.	Poucos rótulos	Robus. a envenen.
[Zhao et al. 2019]	✓	✗	✗	✗
[Li et al. 2021b]	✓	✗	✗	✗
[Zhang et al. 2021]	✓	✗	✗	✗
[Lv et al. 2022]	✓	✗	✗	✗
[Tian et al. 2021]	✓	✗	✗	✗
[Li et al. 2022a]	✗	✗	✗	✓
[Dao et al. 2022b]	✗	✗	✗	✓
[Yin et al. 2022]	✓	✓	✗	✓
[Neto et al. 2022]	✓	✗	✗	✗
[Su and Qu 2022]	✓	✗	✗	✗
[Chen et al. 2022]	✓	✗	✗	✗
[Ali et al. 2023]	✓	✗	✗	✗
[Liu et al. 2023]	✓	✗	✓	✗
Nossa proposta	✓	✓	✓	✓

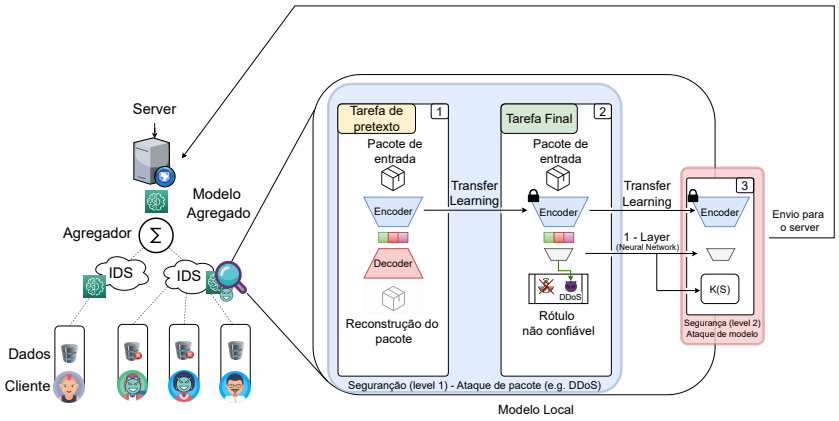

Figura 1. Nossa proposta para detectar ataques à rede em FL inclui dois níveis de segurança: o primeiro (quadrado azul) usa o modelo UL em duas etapas. Primeiro, treinamos um autoencoder para obter os pesos da função codificadora. Depois, essa representação é usada para treinar o classificador de ataques, com os pesos do codificador congelados (icone de cadeado na figura). No segundo nível (quadrado vermelho), avaliamos a aderência dos dados ao modelo para identificar clientes maliciosos utilizando o quantificador $K(\mathcal{S})$.

3. Nossa Proposta

3.1. Tarefa de pretexto

Aprendizado não supervisionado (UL) refere-se a métodos de aprendizado que não necessitam de rótulos anotados por humanos. Entre as abordagens de UL, um caso interessante ocorre quando o modelo gera um pseudo-rótulo para cada dado de entrada, denominado aprendizado auto-supervisionado. Por exemplo, na arquitetura de autoencoder, o pseudo-rótulo é igual ao dado de entrada.

Formalmente, seja o conjunto $\mathcal{O} = \{(\boldsymbol{o}_i)\}_{i=1}^{v}$, com $\boldsymbol{o}_i \in \mathbb{R}^m$, definido em um espaço de características m-dimensional com v elementos. O conjunto $\mathcal{O}$ é denominado *espaço de características original*, e a função de rede neural (também chamada de *encoder*) $e(.\,, \boldsymbol{\theta}) : \mathcal{O} \rightarrow \mathcal{Z}$ é parametrizada por um conjunto de pesos $\boldsymbol{\theta}$, onde $\mathcal{Z}$ é o *espaço de características latente*, de modo que $\mathcal{Z} = \{(\boldsymbol{z}_i)\}_{i=1}^{v}$, com $\boldsymbol{z}_i = e(\boldsymbol{o}_i, \boldsymbol{\theta}) \in \mathbb{R}^n$, $\boldsymbol{z}_i \in \mathbb{R}^n$ e $m > n$ (*dimensão do espaço de características latente*).

Similarmente, a função decodificadora (também chamada de *decoder*) pode ser definida como a função inversa do codificador $e^{-1}(.\,, \boldsymbol{\theta}') : \mathcal{Z} \rightarrow \mathcal{O}$, onde $\boldsymbol{\theta}'$ é um conjunto de pesos para o decodificador. Assim, os pesos ótimos para o autoencoder podem ser estimados minimizando a função de erro quadrático médio $\frac{1}{v}\sum_{i=1}^{v} \|\boldsymbol{o}_i - \boldsymbol{o}'_i\|_2^2$, onde $\boldsymbol{o}'_i = e^{-1}(e(\boldsymbol{o}_i\,, \boldsymbol{\theta})\,, \boldsymbol{\theta}')$ é o pseudo-rótulo associado a $\boldsymbol{o}_i$.

Note que, treinamos nosso modelo local para tarefas de pretexto de aprendizado auto-supervisionado, ou seja, não utilizamos nenhum rótulo (apenas pseudo-rótulos) nesta etapa de treinamento. Nesta fase, estamos interessados em obter uma nova representação dos dados independente de rótulos. Ao final deste treinamento, o modelo $e(.\,, \boldsymbol{\theta})$ mapeia os dados de entrada para um novo espaço de representação $\mathcal{Z}$. Para tarefas de reconstrução, o autoencoder codifica os dados para extrair suas características mais relevantes e os projeta em $\mathcal{O}$. Por último, observe que esta etapa é realizada apenas no cliente (isto é, no

modelo local).

3.2. Tarefa Final

Os métodos baseados em tarefas de pretexto permitem o aprendizado de representações de uso geral, possibilitando o *fine-tuning* supervisionado para otimizar o desempenho em tarefas finais. Após o treinamento da tarefa de pretexto, utilizamos o codificador para extrair uma nova representação da entrada do modelo local, congelando seus pesos para o restante do treinamento. O módulo de projeção é descartado e substituído por uma rede neural de camada única, responsável pela classificação de ataques à rede. Assim, a tarefa final se beneficia dos pesos previamente estimados em um ambiente auto-supervisionado.

Nós utilizamos um modelo de aprendizado supervisionado com o objetivo de mapear os vetores de entradas do conjunto $\mathcal{O}$ para o rótulo de saída correspondente. Formalmente, buscamos encontrar uma função $\ell : \mathcal{O} \rightarrow \mathcal{Y}$, parametrizada por pesos Ω, tal que $\ell(\boldsymbol{o_i}; \Omega)$ preveja o rótulo de saída $\boldsymbol{y}$ para cada $\boldsymbol{o_i} \in \mathcal{O}$. O processo de treinamento envolve minimizar a discrepância entre a saída prevista $\ell(\boldsymbol{o_i}; \Omega)$ e a saída real $\boldsymbol{y}$.

Baseado em [Barros et al. 2022, Barros et al. 2024], utilizamos redes neurais siamesas (SNNs) para medir similaridade ou distância entre entradas. Sua integração em frameworks de meta-aprendizado [Finn et al. 2017] (por exemplo, com foco em rápida adaptação a novas tarefas e objetivos específicos como aprendizado de representações) pode melhorar significativamente a eficiência do aprendizado em ambientes dinâmicos [Li et al. 2021a].

Em nosso framework de meta-aprendizado, uma SNN processa pares do *espaço de características* $\mathcal{X}$, denotados como $(\boldsymbol{o}_i, \boldsymbol{o}_j)$, e os transforma em representações latentes $(\boldsymbol{z}_i, \boldsymbol{z}_j)$ usando o codificador f_Θ. Um par $(\boldsymbol{o}_i, \boldsymbol{o}_j)$ é considerado similar se compartilhar o mesmo rótulo, ou seja, $\ell(\boldsymbol{o}_i) = \ell(\boldsymbol{o}_j)$. Caso contrário, são considerados dissimilares se seus rótulos forem diferentes, ou seja, $\ell(\boldsymbol{o}_i) \neq \ell(\boldsymbol{o}_j)$. Neste modelo definimos um espaço de representação conhecido como *Espaço de Similaridade* ou *S-space*, construído a partir do produto cartesiano $\mathcal{O} \times \mathcal{O}$, conforme descrito em [Barros et al. 2022]. Seja f^S a função $f^S : \mathcal{O} \times \mathcal{O} \rightarrow \mathcal{S}$, onde $\mathcal{S} = \{\boldsymbol{s}_{ij}\}$ representa o espaço de similaridade para cada $\boldsymbol{o_i}, \boldsymbol{o_j} \in \mathcal{O}$. Aplicando f^S a $(\boldsymbol{o}_i, \boldsymbol{o}_j)$, definimos o vetor de similaridade $\boldsymbol{s}_{ij}$ como

$$\boldsymbol{s}_{ij} = |f_\Theta(\boldsymbol{o}_i) - f_\Theta(\boldsymbol{o}_j)| = |\boldsymbol{z}_i - \boldsymbol{z}_j| = \left(\left|z_i^1 - z_j^1\right|, \left|z_i^2 - z_j^2\right|, \ldots, \left|z_i^n - z_j^n\right|\right), \quad (1)$$

onde o vetor de similaridade $\boldsymbol{s}_{ij}$ tem a mesma dimensionalidade que as representações latentes $\boldsymbol{z}_i$ e $\boldsymbol{z}_j$, e z_i^n denota a n-ésima característica do i-ésimo ponto de dados no espaço latente $\mathcal{Z}$.

Com base no *S-space*, definimos marcadores para quantificar a similaridade entre pares de entrada. Temos dois conjuntos de marcadores: $\mathcal{M}^+$ para similaridade e $\mathcal{M}^-$ para dissimilaridade, onde o conjunto completo de marcadores é $\mathcal{M} = \mathcal{M}^+ \cup \mathcal{M}^-$.

Para um par $(\boldsymbol{o}_i, \boldsymbol{o}_j)$ e o vetor de similaridade correspondente $\boldsymbol{s}_{ij}$, a proximidade a um marcador de similaridade $\boldsymbol{\mu}^+ \in \mathcal{M}^+$ indica um maior grau de similaridade entre $\boldsymbol{o}_i$ e $\boldsymbol{o}_j$. Por outro lado, a proximidade a um marcador de dissimilaridade $\boldsymbol{\mu}^- \in \mathcal{M}^-$ indica maior dissimilaridade. Esses marcadores são vetores com a mesma dimensão de $\boldsymbol{s}_{ij}$, e suas posições são ajustadas dinamicamente durante a otimização para representar efetivamente similaridade ou dissimilaridade. Para quantificar a similaridade entre $\boldsymbol{s}_{ij}$ e

um marcador $\boldsymbol{\mu}_m \in \mathcal{M}$, utilizamos uma distribuição de Cauchy discreta como função kernel [Barros et al. 2024]

$$q_{ij}^m = \frac{(1 + \|\boldsymbol{s}_{ij} - \boldsymbol{\mu}_m\|_2^2)^{-1}}{\sum_{\boldsymbol{\mu}_{m'} \in \mathcal{M}} (1 + \|\boldsymbol{s}_{ij} - \boldsymbol{\mu}_{m'}\|_2^2)^{-1}}, \tag{2}$$

onde q_{ij}^m representa o valor de similaridade (ou dissimilaridade) do par $(\boldsymbol{o}_i, \boldsymbol{o}_j)$ em relação ao marcador $\boldsymbol{\mu}_m$.

Definimos as probabilidades cumulativas de similaridade e dissimilaridade como $q_{ij}^+ = \sum_p q_{ij}^p$, para todos $\boldsymbol{\mu}_p \in \mathcal{M}^+$, e $q_{ij}^- = \sum_n q_{ij}^n$, para todos $\boldsymbol{\mu}_n \in \mathcal{M}^-$. Por construção, q_{ij}^+ representa a probabilidade de que o par $(\boldsymbol{o}_i, \boldsymbol{o}_j)$ seja similar (compartilhe o mesmo rótulo), enquanto q_{ij}^- indica a probabilidade de que o par seja dissimilar (tenha rótulos diferentes). Como $\mathcal{M}^+$ e $\mathcal{M}^-$ são conjuntos disjuntos, segue-se que $q_{ij}^+ + q_{ij}^- = 1$.

O processo de otimização para o Espaço de Similaridade (*S-space*) envolve ajustar simultaneamente as posições dos marcadores em $\mathcal{M}$ e atualizar os parâmetros Θ das funções do codificador. A função de perda associada é definida como $J(\mathcal{X} \times \mathcal{X}) = \sum_i \sum_j \left[u_{ij} \log q_{ij}^+ + (1 - u_{ij}) \log q_{ij}^-\right]$, onde $u_{ij} = 1$ se $\boldsymbol{o}_i$ e $\boldsymbol{o}_j$ têm o mesmo rótulo, e $u_{ij} = 0$ caso contrário. Por fim, utilizamos um algoritmo de KNN no espaço de representação $\mathcal{Z}$ para classificar as entradas.

Dessa forma, mapeamos a codificação encontrada pelo codificador em um vetor de probabilidade discreta que quantifica a chance de que os dados de entrada analisados sejam resultado de um ataque à rede (por exemplo, DDoS). Apenas nesta etapa utilizamos o rótulo associado à entrada do modelo para treinar o modelo local.

Uma das características de nossa abordagem é considerar que, além do ataque à rede (ataque ao sistema), a federação está suscetível a um ataque ao modelo (ataque aos rótulos). Logo, na segunda etapa, nosso framework considera que o rótulo usado no treinamento do classificador de DDoS não é confiável, ou seja, consideramos que esse rótulo pode ser adulterado para prejudicar a federação. Nossa proposta pode ser ilustrada na Figura 1.

3.3. Justificativa do Framework

Os autores em [Bansal et al. 2020] propuseram um limite superior para o *gap* de generalização em modelos de redes neurais para tarefas de classificação, que pode ser quantificado em três partes: *Gap de Robustez*, *Gap de Racionalidade* e *Gap de Memorização*. Tipicamente em redes neurais, os *gaps* de robustez e racionalidade são pequenos [Bansal et al. 2020]. Um *gap* de robustez pequeno implica que a adição de uma quantidade limitada de rótulos errados causa uma degradação mínima no desempenho. Por outro lado, o pequeno *gap* de racionalidade sugere que obter um rótulo errado não é melhor do que não obter nenhum rótulo.

No entanto, os classificadores modernos são altamente superparametrizados, logo esses modelos conseguem facilmente ajustar um rotulamento aleatório dos dados de treinamento e, consequentemente, apresentam um alto *gap* de memorização, como pode ser observado em [Zhang et al. 2017]. Para contornar esse problema, [Bansal et al. 2020] apresentaram evidências de que o gap de memorização é negligenciável se o classificador

tiver baixa complexidade, independentemente da complexidade da representação. O *gap* de memorização tende a zero se o classificador for subparametrizado.

Nós hipotetizamos que nossa abordagem mitiga o problema de alto *gap* de memorização e, consequentemente, o ataque de envenenamento de rótulos em ambientes de FL. Embora nosso modelo auto-supervisionado possa ser altamente parametrizado, o ataque de envenenamento de rótulos não afeta o desempenho do modelo. Além disso, treinamos um classificador subparametrizado que tende a minimizar o *gap* de memorização. Por fim, estimamos a quantificação da aderência dos dados associada ao classificador subparametrizado para detectar ataques de envenenamento de rótulos.

3.4. Quantificador de assertividade

Definição 1. ***(Espaço Latente Ótimo)*** *Seja $\boldsymbol{o}_i, \boldsymbol{o}_j \in \mathcal{O}$ e uma função de representação latente $f_\Theta : \mathcal{O} \to \mathcal{Z}$. A transformação f_Θ gera um espaço latente ótimo $\mathcal{Z}$ quando o valor esperado $\mathbb{E}[||\boldsymbol{s}_{ij}||_2] = 0 \implies \ell(\boldsymbol{o}_i) = \ell(\boldsymbol{o}_j)$, onde a função $\ell : \mathcal{O} \to \mathcal{Y}$ mapeia um exemplo não rotulado $\boldsymbol{o}_i$ para seu respectivo rótulo y_i.*

Suposição 1. *Considere a função de risco de similaridade de classificação incorreta $\mathcal{R}^+(\mathcal{S})$ para o modelo $\mathcal{S}$. Para dois modelos de espaço latente ótimo com dois marcadores $\mathcal{M}^+ = \{\mu^+\}$ e $\mathcal{M}^- = \{\mu^-\}$, $\mathcal{S}_1$ e $\mathcal{S}_2$; se $\mathcal{R}^+(\mathcal{S}_1) < \mathcal{R}^+(\mathcal{S}_2)$, então assume-se que $\mathbb{E}_{o_i,o_j}[q_{ij}^+]_1 > \mathbb{E}_{o_i,o_j}[q_{ij}^+]_2$ para pares similares (o_i, o_j), onde $[q_{ij}^+]_*$ representa a saída de similaridade para o modelo $\mathcal{S}_*$.*

> Como evidenciado em [Barros et al. 2024], nosso modelo reduz o risco de classificação incorreta ao induzir q^+ a 1 quando a performance é alta (ver Lema C.1. em [Barros et al. 2024]). No *espaço latente ótimo*, pares similares colapsam ($\|s_{ij}\|_2 = 0$) e a separação inter-classe é maximizada, conforme a Definição 1. Assim, assumimos que, para dois modelos $\mathcal{S}_1$ e $\mathcal{S}_2$ satisfazendo a Definição 1, se $\mathcal{R}^+(\mathcal{S}_1) < \mathcal{R}^+(\mathcal{S}_2)$, então $\mathbb{E}[q^+]_1 > \mathbb{E}[q^+]_2$, pois um menor risco de classificação incorreta implica em uma maior precisão na similaridade.

Lema 1. *Considere a função de risco de similaridade de classificação incorreta $\mathcal{R}^+(\mathcal{S})$ para o modelo $\mathcal{S}$. Para dois modelos de espaço latente ótimo, $\mathcal{S}_1$ e $\mathcal{S}_2$, se $\mathcal{R}^+(\mathcal{S}_1) < \mathcal{R}^+(\mathcal{S}_2)$, então segue que $K(\mathcal{S}_1) > K(\mathcal{S}_2)$, onde $K(\mathcal{S}) = 1 - \frac{1+|\boldsymbol{\mu}^+|_2^2}{1+\|\boldsymbol{\mu}^-\|_2^2}$, $\mathcal{M}^+ = \{\mu^+\}$ e $\mathcal{M}^- = \{\mu^-\}$.*

Demonstração. Pela Definição 1, para o Espaço Latente Ótimo, temos $\mathbb{E}[\|\boldsymbol{s}_{ij}\|] = 0$, logo segue que $q_{ij}^m = \frac{\left(1+\|\boldsymbol{s}_{ij}-\boldsymbol{\mu}_m\|_2^2\right)^{-1}}{\sum \boldsymbol{\mu}_{m'} \in \mathcal{M}\left(1+\|\boldsymbol{s}_{ij}-\boldsymbol{\mu}_{m'}\|_2^2\right)^{-1}}$. No caso ótimo, quando $\mathbb{E}[|\boldsymbol{s}_{ij}|] = 0$, isso se simplifica para

$$q_{ij}^m = \frac{\left(1 + \|\boldsymbol{\mu}_m\|_2^2\right)^{-1}}{\sum \boldsymbol{\mu}_{m'} \in \mathcal{M}\left(1 + \|\boldsymbol{\mu}_{m'}\|_2^2\right)^{-1}}.$$

Pela Suposição 1, no espaço $\mathcal{S}$ com dois marcadores, $\boldsymbol{\mu}^+$ e $\boldsymbol{\mu}^-$, representando marcadores positivos e negativos, respectivamente, podemos expressar a relação entre os modelos $\mathcal{S}_1$ e $\mathcal{S}_2$ como $\frac{1}{1+\frac{(1+\|\boldsymbol{\mu}_1^+\|_2^2)^{-1}}{(1+\|\boldsymbol{\mu}_1^-\|_2^2)^{-1}}} > \frac{1}{1+\frac{(1+\|\boldsymbol{\mu}_2^+\|_2^2)^{-1}}{(1+\|\boldsymbol{\mu}_2^-\|_2^2)^{-1}}}$.

Reescrevendo essa desigualdade e multiplicando pelos denominadores (que são positivos) temos que $(1 + \|\boldsymbol{\mu}_1^+\|_2^2)(1 + \|\boldsymbol{\mu}_2^-\|_2^2) > (1 + \|\boldsymbol{\mu}_2^+\|_2^2)(1 + \|\boldsymbol{\mu}_1^-\|_2^2)$. Dividindo ambos os lados por $(1 + \|\boldsymbol{\mu}_1^+\|_2^2)(1 + \|\boldsymbol{\mu}_2^+\|_2^2)$ e subtraindo essas frações de 1, obtemos

$$1 - \frac{1 + \|\boldsymbol{\mu}_2^-\|_2^2}{1 + \|\boldsymbol{\mu}_2^+\|_2^2} < 1 - \frac{1 + \|\boldsymbol{\mu}_1^-\|_2^2}{1 + \|\boldsymbol{\mu}_1^+\|_2^2}.$$

Pela definição de $K(\mathcal{S})$, isso implica que $K(\mathcal{S}_1) > K(\mathcal{S}_2)$. Assim, o lema está provado. □

Nesta seção, introduzimos um quantificador de assertividade $K(\mathcal{S})$ para nosso modelo federado, fundamentado na definição de espaço latente ótimo e na relação entre risco de classificação incorreta e precisão de similaridade. O objetivo é estabelecer um critério quantitativo para avaliar a confiabilidade das previsões do modelo em diferentes cenários.

3.5. Modelo de Ataque FL: Ataque por Envenenamento de Rótulos

Considere N usuários $\{u_1, \cdots, u_N\}$, todos desejando treinar um modelo de ML utilizando seus respectivos conjuntos de dados $\{\mathcal{O}_1, \cdots, \mathcal{O}_N\}$. O método tradicional de treinamento em ML consiste em agrupar todos os dados no conjunto $\mathcal{O} = \mathcal{O}_1 \cup \cdots \cup \mathcal{O}_N$ para treinar um modelo $T_{\mathcal{O}}$. Um sistema de FL é um processo em que os proprietários dos dados treinam um modelo colaborativamente, denominado T_F. Nesse processo, qualquer proprietário de dados u_i não expõe seus dados $\mathcal{O}_i$ para os outros. De maneira geral, o processo de treinamento FL inclui os seguintes três passos:

- Passo 1 (Inicialização da Tarefa): O sistema inicializa os modelos locais $T_{\mathcal{O}_i}$ e define todos os valores necessários dos hiperparâmetros (por exemplo, taxa de aprendizado) para iniciar a tarefa de aprendizado.
- Passo 2 (Treinamento e Atualização do Modelo Local): Cada usuário u_i usa seus dados locais $\mathcal{O}_i$ e seu dispositivo para atualizar o modelo local $T_{\mathcal{O}_i}$. O objetivo do usuário é encontrar os parâmetros ótimos para o modelo local que minimizem a função de perda.
- Passo 3 (Agregação e Atualização do Modelo): O servidor agrega os modelos locais dos participantes selecionados e envia as atualizações de volta para os proprietários dos dados.

Os passos 2 e 3 são repetidos até que a função de perda agregada convirja ou atinja uma métrica de treinamento desejável. A agregação do modelo pode ser implementada de várias maneiras; por exemplo, o FedAvg [McMahan et al. 2017] realiza a "média aritmética" dos pesos dos modelos locais para obter o modelo agregado no servidor.

Neste trabalho, propomos uma versão modificada da abordagem FedAvg para estimar o modelo global federado Θ_F como:

$$\Theta_F \leftarrow \Theta_F + \eta \sum_{i=1}^{N_{sel}} \left[\mathbb{1}[K(\mathcal{S}_i) \geq t_l] K(\mathcal{S}_i)(\Theta_{\mathcal{O}_i}^* - \Theta_F)\right], \tag{3}$$

onde $\Theta_{\mathcal{O}_i}^*$ é uma estimativa de máxima verossimilhança a posteriori do cliente i; N_{sel} é o número de clientes selecionados na rodada de treinamento; η é a taxa de aprendizado;

$\mathbb{1}$ é a função indicadora, ou seja, $\mathbb{1}[.] = 1$ se [.] for verdadeiro e 0 caso contrário; t_l são limiares, e $K(\mathcal{S}_i)$ é o quantificador definido no Lema 1[1].

Em um ataque por envenenamento federado, um usuário malicioso envia um modelo local ao servidor projetado para degradar o modelo agregado. Mais especificamente, o **ataque federado por envenenamento de rótulos** considera ataques realizados por usuários maliciosos que treinam modelos locais com rótulos manipulados para degradar o modelo agregado. Nesse caso, o ataque afeta diretamente o modelo agregado, independentemente da aplicação. Essas suposições ajudam a avaliar a robustez do modelo contra ataques extremos. Essa análise pode ser benéfica para aplicações que requerem níveis específicos de garantia quanto ao desempenho do ambiente FL.

4. Metodologia

Os ataques DDoS representam um grande desafio, exigindo estratégias eficazes de detecção e mitigação. Neste trabalho, utilizamos dois conjuntos de dados para avaliação de ataques:

CSE-CIC-IDS2018 [Sharafaldin et al. 2018] contém tráfego benigno e ataques DDoS, incluindo fluxos rotulados com informações como IPs, portas, protocolos e timestamps. Além disso, foram extraídas 80 características de tráfego relevantes;

NF-UNSW-NB15-V2 [Sarhan et al. 2022] abrange dez categorias de ataques, com 49 características extraídas de payloads e cabeçalhos, permitindo uma análise aprofundada do tráfego malicioso.

Utilizamos um *autoencoder* auto-supervisionado, com pesos inicializados a partir de uma distribuição normal e vieses de média $0,5$ e desvio padrão 10^{-2} [Koch et al. 2015]. O conjunto de dados foi dividido em treinamento (80%), validação (10%) e teste (10%). O *encoder* possui arquitetura m–2048–512, enquanto o *decoder* é simétrico. O módulo de projeção segue a configuração 256-n, onde m é o número de características e $n = 64$. Todas as camadas são totalmente conectadas, com ativação ReLU.

O treinamento foi realizado com Gradiente Estocástico Descendente em minibatches (SGD), taxa de aprendizado de 10^{-2} e hiperparâmetros ajustados conforme as propostas originais. Por fim, para otimizar os hiperparametros do nosso modelo, nós utilizamos Otimização Bayesiana com seis pontos iniciais aleatórios, seguidos de 20 rodadas de otimização, obtendo o melhor desempenho quando os marcadores de similaridade e dissimilaridade são iguais a 2 e $t_l = 0,51$. Mantemos esses valores em todos os experimentos subsequentes.

4.1. Cenário de rede

Utilizamos o framework Flower para desenvolver soluções em FL, com distribuição não-IID dos dados entre os usuários. Os dados de treinamento foram distribuídos de forma não uniforme entre os clientes (desequilíbrio de rótulos baseado na quantidade) [Li et al. 2022b]. Cada cliente treinou seu modelo localmente, enquanto o servidor

[1]Para manter a clareza na notação, não representamos os lotes locais do processo de aprendizado federado na eq.(3), embora o processo real ocorra a cada certo número de lotes locais.

central agregou as atualizações. O modelo final foi avaliado no conjunto de teste, garantindo conformidade com os princípios de privacidade e aprendizado descentralizado.

O experimento envolveu um servidor e 50 clientes, sendo treinado em uma GPU NVIDIA Quadro RTX 6000 (24 GB) por 100 épocas. A cada rodada, cinco clientes eram selecionados para treinar seus modelos locais, que posteriormente foram agregados no servidor central. Para avaliar ataques de envenenamento de rótulos, consideramos ruído não correlacionado, onde cada rótulo tinha igual probabilidade de ser corrompido. Definimos ρ como a fração de clientes maliciosos e $\gamma = 40\%$ como a taxa de corrupção dos rótulos [Hendrycks et al. 2018]. Para ataques DDoS, seguimos [Dao et al. 2022a], assumindo uma alta porcentagem de usuários maliciosos para simular cenários de estresse.

5. Resultados Experimentais

Nesta seção, apresentamos os resultados do nosso experimento de FL, focado em avaliar a vulnerabilidade de dois conjuntos de dados a ataques de envenenamento de rótulos. Dessa forma, avaliamos as seguintes abordagens para demonstrar a eficácia do nosso método proposto: Median [Yin et al. 2018], Krum [Blanchard et al. 2017], Geometric [Pillutla et al. 2022], Trimmed [Fang et al. 2020] e FedEqual [Chen et al. 2021].

Para realizar um experimento mais realista, assumimos que o administrador da rede não consegue rotular todos os dados de treinamento no sistema. A coleta e anotação de dados geralmente são processos caros e tediosos. Consequentemente, nem sempre temos um grande volume de dados rotulados para treinar abordagens supervisionadas. Em nosso experimento, realizamos uma amostragem aleatória de 5% de cada conjunto de dados de treinamento dos clientes para usar como dados rotulados, e todos os demais dados de treinamento restantes foram considerados não rotulados para treinar a abordagem auto-supervisionada.

Para a avaliação quantitativa, comparamos o desempenho de cada abordagem com o método *Oracle*, que envolve o treinamento de uma rede neural convencional com a função de perda *softmax cross–entropy*. O método *Oracle* se beneficia do acesso a todos os dados não corrompidos e segue uma abordagem centralizada, sem amostras envenenadas.

Para avaliar o impacto de um ataque de envenenamento ao modelo, utilizamos o F1-score do modelo atacado, comparando-o com o F1-score do modelo *Oracle*. A partir dessa comparação, calculamos o erro de degradação, denotado como $d_{\text{error}} = 1 - \frac{\text{F1}}{\text{F1}_{\text{Oracle}}}$ onde F1 representa o F1-score do modelo atacado e $\text{F1}_{\text{Oracle}}$ representa o F1-score do modelo Oracle. Essa métrica estima a degradação do modelo (valores mais altos indicam maior degradação).

Os resultados, apresentados na Tabela 2, demonstram a eficácia dos diferentes algoritmos de FL em melhorar o desempenho dos sistemas de detecção de DDoS. Além disso, o F1-Score do modelo Oracle fornece um ponto de referência para avaliar a eficácia dos algoritmos. Para o conjunto de dados **CSE-CIC-IDS2018** (I), com $\rho = 0$, observamos que a nossa abordagem alcançou o melhor resultado (menor erro de degradação) no conjunto de dados analisado, superando todos os outros algoritmos avaliados. Para referência, o modelo Oracle alcançou um F1-Score de $0,982$.

Na presença de clientes maliciosos, observamos que o desempenho das abordagens é degradado em comparação à nossa proposta. Para o cenário com 20% de clientes

Tabela 2. Avaliação comparativa para diferentes proporções de clientes ruidosos (ρ), considerando 5% dos dados de treinamento rotulados. Os valores de F1-score são calculados tanto para o modelo atacado quanto para o modelo oráculo. O erro de degradação é calculado como um menos a razão entre os F1-scores. Valores mais altos indicam maior degradação do modelo em comparação ao modelo Oracle. Por fim, os melhores resultados estão em negrito

Conjunto de dados	Proposta	$d_{\text{error}} = 1 - \text{F1}/\text{F1}_{\text{Oracle}}$ ($\downarrow$)			
		$\rho = 0$	$\rho = 0,2$	$\rho = 0,4$	$\rho = 0,6$
CSE-CIC-IDS2018	Median	6,7%	16,7%	36,6%	48,3%
	Krum	8,9%	16,0%	33,2%	50,1%
	Geometric	7,8%	10,8%	30,3%	46,9%
	Trimmed	9,0%	11,4%	35,8%	57,7%
	FedEqual	6,3%	**10,0%**	27,1%	52,5%
	Nossa proposta	**4,9%**	10,6%	**19,2%**	**30,8%**
	Oracle (F1-Score)	0,982	0,967	0,955	0,941
NF-UNSW-NB15-V2	Median	6,2%	14,3%	21,3%	49,2%
	Krum	7,1%	15,4%	25,7%	42,3%
	Geometric	6,7%	10,6%	20,0%	36,4%
	Trimmed	5,9%	12,1%	28,1%	59,2%
	FedEqual	6,9%	9,4%	**18,7%**	31,6%
	Nossa proposta	**5,4%**	**8,2%**	19,8%	**28,6%**
	Oracle (F1-Score)	0,921	0,910	0,904	0,891

maliciosos ($\rho = 0,2$), nossa abordagem obtém o segundo melhor resultado, apresentando uma diferença de $0,6\%$ de erro de degradação da melhor proposta (para referência, o F1-Score do Oracle foi de 0.967). Entretanto, observamos que nossa abordagem obtém o melhor resultado ao analisarmos cenários extremos com maior proporção de usuários maliciosos. Por exemplo, para o cenário onde $\rho = 0,6$, observamos que nossa abordagem apresenta $30,8\%$ de erro de degradação (o Oracle alcançou $0,941$ no F1-Score), resultando em uma diferença de $11,7\%$ em relação ao FedEqual (segundo melhor resultado).

Por fim, no experimento utilizando o conjunto de dados **NF-UNSW-NB15-V2** (II), verificamos que, no caso sem usuários maliciosos ($\rho = 0$), nossa proposta obteve o melhor resultado. Especificamente, nossa abordagem alcança um erro de degradação de 8,2%, 19,7% e 28,6% para as respectivas proporções de clientes maliciosos, obtendo o melhor resultado em dois dos 3 cenários analisados neste conjunto de dados. Em comparação, outras técnicas de agregação exibem erros de degradação maiores, com **Median**, **Krum**, **Geometric**, **Trimmed** e **FedEqual** variando de $14,3\%$ a $49,2\%$, $15,4\%$ a $42,3\%$, $10,6\%$ a $36,4\%$, $12,1\%$ a $59,2\%$ e $6,9\%$ a $31,6\%$, respectivamente. Os resultados experimentais destacam a eficácia de nosso método proposto em lidar com ataques de envenenamento de rótulos, superando consistentemente outras técnicas de agregação.

6. Conclusão

Propomos, neste trabalho, um novo modelo distribuído para detectar ataques em redes. Nossa proposta apresenta evidências de que, além de detectar ataques DDoS, ela é robusta contra ataques ao modelo, garantindo assim dois níveis de segurança. Projetamos nossa abordagem especificamente para resistir a ataques de envenenamento de rótulos. Nosso

framework alcança um desempenho superior quando comparado às técnicas do estado da arte. Essa conclusão é especialmente evidente em cenários com uma fração significativa de usuários maliciosos ($\rho \geq 0,2$).

Por fim, vale ressaltar que o modelo SSN congela os pesos do codificador, resultando em um modelo de menor capacidade. Como discutido neste artigo, quantificadores que consideram o equilíbrio entre a capacidade do modelo e o erro de generalização mostram-se relevantes para o treinamento de modelos em FL. Portanto, investigar como quantificar o efeito da Navalha de Occam em um ambiente distribuído e heterogêneo (por exemplo, FL) é uma direção promissora para pesquisas futuras.

Agradecimentos

Este trabalho foi parcialmente financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) (grant #2023/00721-1), Conselho Nacional de Desenvolvimento Científico e Tecnológico (grant #312682/2021-2), Fundação de apoio da UFMG – Fundep – (grant #29271) e Fundação de Amparo a Pesquisa do Estado de Minas Gerais (grant #APQ-00426-22).

Referências

Ali, M. N., Imran, M., din, M. S. u., and Kim, B.-S. (2023). Low rate ddos detection using weighted federated learning in sdn control plane in iot network. *Applied Sciences*, 13(3).

Bansal, Y. et al. (2020). For self-supervised learning, rationality implies generalization, provably. In *International Conference on Learning Representations (ICLR)*.

Barros, P. H., Chagas, E. T., Oliveira, L. B., Queiroz, F., and Ramos, H. S. (2022). Malware-smell: A zero-shot learning strategy for detecting zero-day vulnerabilities. *Computers & Security*, 120:102785.

Barros, P. H., Murai, F., Houmansadr, A., Frery, A. C., and Ramos, H. S. (2024). Variational inference in similarity spaces: A bayesian approach to personalized federated learning. In *NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty*.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30.

Chen, J., Guo, Q., Fu, Z., Shang, Q., Ma, H., and Wu, D. (2022). Campus network intrusion detection based on federated learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Chen, L.-Y., Chiu, T.-C., Pang, A.-C., and Cheng, L.-C. (2021). Fedequal: Defending model poisoning attacks in heterogeneous federated learning. In *IEEE Global Communications Conference (GLOBECOM)*.

Dao, N.-N. et al. (2022a). Securing heterogeneous IoT with intelligent DDoS attack behavior learning. *IEEE Systems Journal*, 16(2):1974–1983.

Dao, N.-N., Phan, T. V., Sa'ad, U., Kim, J., Bauschert, T., Do, D.-T., and Cho, S. (2022b). Securing heterogeneous IoT with intelligent DDoS attack behavior learning. *IEEE Systems Journal*, 16(2):1974–1983.

Fang, M., Cao, X., Jia, J., and Gong, N. (2020). Local model poisoning attacks to Byzantine-Robust federated learning. In *29th USENIX Security Symposium*, pages 1605–1622. USENIX Association.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. (2018). Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in Neural Information Processing Systems*, volume 31.

Issa, W., Moustafa, N., Turnbull, B., Sohrabi, N., and Tari, Z. (2023). Blockchain-based federated learning for securing internet of things: A comprehensive survey. *ACM Computing Surveys*, 55(9).

Koch, G. et al. (2015). Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2.

Lavaur, L., Pahl, M.-O., Busnel, Y., and Autrel, F. (2022). The evolution of federated learning-based intrusion detection and mitigation: A survey. *IEEE Transactions on Network and Service Management*, pages 2309–2332.

Li, C., Niu, D., Jiang, B., Zuo, X., and Yang, J. (2021a). Meta-har: Federated representation learning for human activity recognition. In *Proceedings of the Web Conference 2021*, WWW '21, page 912–922.

Li, J. et al. (2022a). FLEAM: A federated learning empowered architecture to mitigate DDoS in industrial IoT. *IEEE Transactions on Industrial Informatics*, 18(6):4059–4068.

Li, J., Zhang, Z., Li, Y., Guo, X., and Li, H. (2021b). FIDS: Detecting DDoS through federated learning based method. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 856–862.

Li, Q., Diao, Y., Chen, Q., and He, B. (2022b). Federated learning on non-iid data silos: An experimental study. In *IEEE International Conference on Data Engineering*.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, pages 50–60.

Liu, Z., Guo, C., Liu, D., and Yin, X. (2023). An asynchronous federated learning arbitration model for low-rate ddos attack detection. *IEEE Access*, 11.

Lv, D., Cheng, X., Zhang, J., Zhang, W., Zhao, W., and Xu, H. (2022). DDoS attack detection based on cnn and federated learning. In *International Conference on Advanced Cloud and Big Data*, pages 236–241.

McMahan, B. et al. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282.

Neto, E. C. P., Dadkhah, S., and Ghorbani, A. A. (2022). Collaborative DDoS detection in distributed multi-tenant iot using federated learning. In *19th Annual International Conference on Privacy, Security & Trust*.

Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., and Vincent Poor, H. (2021). Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3):1622–1658.

Pillutla, K., Kakade, S. M., and Harchaoui, Z. (2022). Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154.

Sarhan, M., Layeghy, S., and Portmann, M. (2022). Towards a standard feature set for network intrusion detection system datasets. *Mob. Netw. Appl.*, 27(1):357–370.

Sharafaldin, I. et al. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *International Conference on Information Systems Security and Privacy (ICISSP)*, pages 108–116.

Su, D. and Qu, Z. (2022). Detection ddos of attacks based on federated learning with digital twin network. In Memmi, G., Yang, B., Kong, L., Zhang, T., and Qiu, M., editors, *Knowledge Science, Engineering and Management*, pages 153–164, Cham.

Tian, Q., Guang, C., Wenchao, C., and Si, W. (2021). A lightweight residual networks framework for DDoS attack classification based on federated learning. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops*, pages 1–6.

Toldinas, J., Venčkauskas, A., Liutkevičius, A., and Morkevičius, N. (2022). Framing network flow for anomaly detection using image recognition and federated learning. *Electronics*, 11(19).

Van Engelen, J. E. and Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine learning*, 109(2):373–440.

Wang, H., Sreenivasan, K., Rajput, S., Vishwakarma, H., Agarwal, S., Sohn, J.-y., Lee, K., and Papailiopoulos, D. (2020). Attack of the tails: Yes, you really can backdoor federated learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the International Conference on Machine Learning*, volume 80, pages 5650–5659.

Yin, Z., Li, K., and Bi, H. (2022). Trusted multi-domain DDoS detection based on federated learning. *Sensors*.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*.

Zhang, J., Yu, P., Qi, L., Liu, S., Zhang, H., and Zhang, J. (2021). FLDDoS: DDoS attack detection model based on federated learning. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 635–642.

Zhao, Y., Chen, J., Wu, D., Teng, J., and Yu, S. (2019). Multi-task network anomaly detection using federated learning. In *Proceedings of the 10th International Symposium on Information and Communication Technology*, SoICT '19, page 273–279.